УДК 004.27

Г.М. Луцький, І.А. Клименко
Національний технічний університет України «Київський політехнічний інститут»
Україна, 03056, м. Київ, проспект Перемоги, 37

ЗМЕНШЕННЯ НАКЛАДНИХ ВИДАТКІВ РЕКОНФІГУРАЦІЇ В РЕКОНФІГУРОВАНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

G.M. Luckiy, I.A. Klymenko
National Technical University of Ukraine "Kyiv Polytechnic Institute"
Ukraine, 03056, Kyiv, Prospect Peremohy 37

REDUCED RECONFIGURATION OVERHEAD IN THE RECONFIGURABLE COMPUTING SYSTEMS

Г.М. Луцкий, И.А. Клименко
Национальный технический университет Украины «Киевский политехнический институт»
Украина, 03056, г. Киев, проспект Победы, 37

СОКРАЩЕНИЕ НАКЛАДНЫХ РАСХОДОВ РЕКОНФИГУРАЦИИ В РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Запропоновано нові засоби реконфігурації, що дозволяють мінімізувати накладні видатки реконфігурації, забезпечуючи підвищення продуктивності динамічно реконфігурованих обчислювальних систем. Запропонований метод повторного використання апаратних ресурсів функціональних блоків, забезпечує інтенсивне прискорення реконфігурації за рахунок видалення всієї непродуктивної складової часу реконфігурації.

Ключові слова: реконфігуровані обчислювальні системи, накладні видатки реконфігурації, прискорення реконфігурації, ПЛІС.

The new means of the reconfiguration is proposed that to minimize overheads reconfiguration and to provide improved performance dynamically reconfigurable systems. The proposed reusing hardware resources method of function blocks provides an intensive acceleration reconfiguration by removing all the overhead part-time reconfiguration.

Keywords: reconfigurable computer systems, reconfiguration overhead, rapid configuration, FPGA.

Предложены новые средства реконфигурации, которые позволяют минимизировать накладные расходы реконфигурации, обеспечивая повышение производительности динамически реконфигурируемых вычислительных систем. Предложенный метод повторного использования аппаратных ресурсов функциональных блоков обеспечивает интенсивнее ускорение реконфигурации за счет удаления всей непродуктивной составляющей времени реконфигурации.

Ключевые слова: реконфигурируемые вычислительные системы, накладные расходы реконфигурации, ускорение реконфигурации, ПЛИС.

Сучасні технології ПЛІС дозволили надати паралельним реконфігурованим обчислювальним системам здатності здійснення реконфігурації архітектури в динамічному режимі, що підвищує ефективність рішення проблеми адаптації структури обчислювальної до структурних та часових вимог вирішуваних задач [1, 2, 3]. При цьому значною проблемою, що негативно впливає на продуктивність обчислень, є накладні видатки реконфігурації, обумовлені непродуктивними витратами часу, продуктивності й енергоресурсів для забезпечення процесу реконфігурації. Супутньою цьому існує проблема обмежень апаратних ресурсів

програмованої елементної бази [2, 4], де критичними параметрами є об'єм внутрішньої пам'яті та швидкодія інтерфейсів ПЛІС. Також, до уваги слід приймати кількісні обмеження логічних ресурсів мікросхеми. Це впливає на ефективність просторових рішень під час розміщення конфігурацій апаратних задач на поверхні реконфігурованої структури. Подолання визначених проблем має високу актуальність в галузі високопродуктивних реконфігурованих обчислень і не може бути ефективно вирішено відомими методами і засобами.

Огляд відомих рішень

Більшість сучасних досліджень в галузі реконфігурованих обчислень, детальний аналіз яких представлений в роботі [4], спрямовані на розробки алгоритмів планування й розміщення задач, що водночас підтримують різні методи зменшення витрат реконфігурації. Це підвищує їх обчислювальну складність та спричиняє додаткові накладні витрати реконфігурації. Всі відомі методи забезпечують прискорення реконфігурації, за рахунок максимально можливого зменшення затримок реконфігурації, частіше за все, ціною надлишкового використання апаратних ресурсів ПЛІС. Це спричиняє великий відсоток відмов виконання задач і додаткові витрати часу й продуктивності. Проблема оптимального зменшення накладних видатків за критеріями забезпечення необхідного часу обчислення та ефективного використання апаратних ресурсів розглянута в роботі [2], але запропоновані засоби орієнтовані на визначену наперед структуру реконфігурованої області та вимагають попередньої структуризації задачі.

Постановка задачі

Вихідний обчислюваний додаток є паралельною програмою зі змішаним типом паралелізму, для опису якого застосовується модель програмування М-програм [5]. М-програма задається макрографом потоку даних *MDG* (*Macro Dataflow Graphs*), у вершинах якого розміщуються макрозадачі (М-задачі), а ребра вказують на залежності між вершинами.

Визначимо час виконання М-задачі як:

$$T = T_{Rconf} + T_{HW} ,$$

де T_{HW} – час обчислення задачі апаратними засобами, T_{Rconf} – час реконфігурації, витрачений безпосередньо на розміщення конфігурації відповідної апаратної задачі на поверхні реконфігурованої області обчислювального модуля, N – кількість вершин графу М-програми. Час T_{Rconf} (1) визначає основний обсяг накладних видатків реконфігурації, що негативно впливають на продуктивність обчислень. Непродуктивна компонента часу реконфігурації визначається тривалістю циклу управління процесом реконфігурації $T_{control}$ – обумовлює втрати продуктивності, часом передавання конфігураційних даних T_{comm} – обумовлює комунікаційні витрати, часу прошивання мікросхеми T_{mapp} – залежить від технологічних особливостей реконфігурованої елементної бази. Тоді проблема прискорення реконфігурації зводиться до рішення наступної формальної задачі зменшення непродуктивних витрат часу:

$$T_{reconf} = \min(T_{control}) + \min(T_{comm}) + T_{mapp} , \quad (1)$$

при цьому зробимо припущення, що інтенсивність прискорення реконфігурації знаходиться в прямій залежності від кількості видалених непродуктивних витрат.

Метод повторного використання апаратних ресурсів функціональних блоків

Подамо вихідну М-програму макрографом $G_M = (V, E)$, де V – множина вершин, що відповідають макрозадачам, а E – множина ребер, що визначають відношення між макрозадачами (рис. 1).

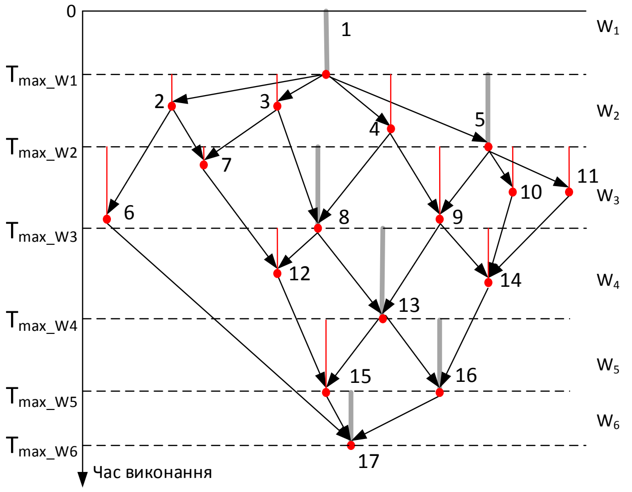


Рис. 1. Макрограф обчислювального алгоритму поданого в ЯПФ

За подання макрографу програми в ярусно-паралельній формі (ЯПФ) зручно використати спосіб відображення задач на структуру реконфігурованої системи за рівнями, коли кожний рівень підграфу послідовно відображується на доступну обчислювальну структуру [5]. Тоді час виконання обчислюваної програми визначається, як сумарний часу виконання найтриваліших задач кожного ярусу, і описується виразом:

$$T_G = \sum_{k=1}^{w} \max \{T_h \middle| h = \overline{1, H_k}\} , \quad (2)$$

де $k = \overline{1, w}$ – номер ярусу, w – кількість ярусів обчислювального алгоритму, $h = \overline{1, H_k}$ – номер вузла на ярусі k, H_k – кількість вузлів на ярусі k. Відповідно до цього, наступна упорядкована множина М-задач:

$$B = \{T_{\max_k} \middle| k = \overline{1, w}\} , \quad (3)$$

де $T_{\max_k}$ – час виконання найтривалішої задачі k-го ярусу, складає найтривалішу взаємозв'язану послідовність задач, яка визначає час виконання програми.

Нехай час виконання послідовності B (3) обчислювальних задач $I_j \mid j = \overline{1, K}$ (де K – кількість типів апаратних задач в межах графу G_M обчислювального алгоритму) виражається як

$$T_B = \sum_{j=1}^{K} (P_j T_j + P_j \Delta T_j) , \quad (4)$$

де P_j – кількість екземплярів, а ΔT_j – час завантаження та конфігурування кожної j-ї апаратної задачі в послідовності обчислень B.

Запропонований метод повторного використання апаратних ресурсів функціональних блоків (МПВ) забезпечує наступний час виконання послідовності задач:

$$T_B^{\text{МПВ}} = \sum_{j=1}^{K}(P_j T_j + \Delta T_j), \qquad (5)$$

Складова $P_j T_j$ у виразі (5) відповідає продуктивному сумарному часу виконання всіх екземплярів апаратної задачі I_j на поверхні реконфігурованої області. Складова ΔT_j відповідає часу її одноразової конфігурації. Показник абсолютного прирощення

$$\Delta T_{\text{МПВ}} = T_B - T_B^{\text{МПВ}} = \sum_{j=1}^{K}(P_j \Delta T_j - \Delta T_j) = \sum_{j=1}^{K}(P_j(1-\Delta T_j)) \qquad (6)$$

зображує, що метод повторного використання апаратних ресурсів функціональних блоків видаляє весь непродуктивний час виконання обчислень, що відповідає часу повторних реконфігурацій послідовності виконуваних задач. З виразу (6) видно, що розмір непродуктивної компоненти часу прямо пропорційно за лінійним законом залежить від часу конфігурування апаратної задачі і кількості екземплярів однотипних задач.

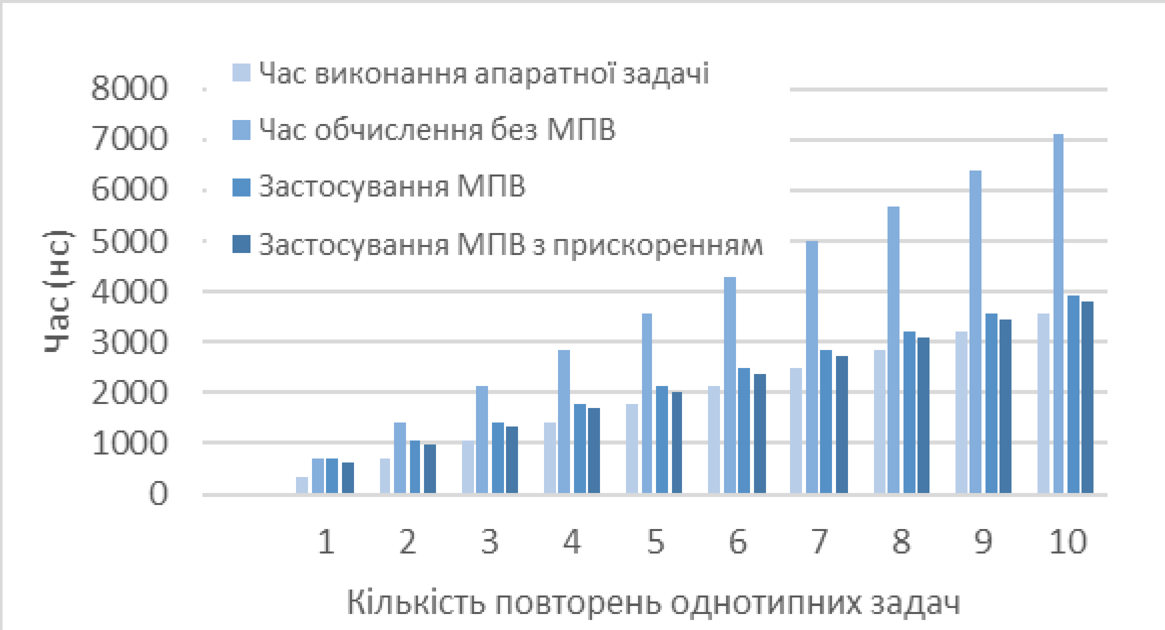


Рис. 2. Залежність швидкодії обчислень від кількості повторів однотипних задач

Застосування швидкодіючої локальної пам’яті в структурі обчислювального модуля дозволяє отримати додаткове прискорення під час застосування МПВ за рахунок зменшення затримки ΔT_j в виразі (5).

На діаграмі (рис. 2) представлені залежності часу виконання послідовності задач від кількості повторів однотипних задач за застосування запропонованого методу повторного використання апаратних ресурсів функціональних блоків. Інтенсивність прискорення збільшується зі збільшенням кількості екземплярів однотипних задач, яка за лінійним законом обґрунтовує обсяг накладних видатків реконфігурації (6).

На підставі виразів (2) та (5), отримаємо математичну модель методу повторного використання ресурсів функціональних блоків для макрографу G_M, поданого в ЯПФ алгоритму.

З виразу (5) виходить, що час виконання будь-якого послідовного алгоритму обчислення визначається як сума часу виконання всіх екземплярів апаратних задач і часу їх одноразової конфігурації:

$$T^{\text{МПВ}} = \sum_{j=1}^{K}(P_j T_j) + \sum_{j=1}^{K}\Delta T_j \,, \tag{7}$$

За розпаралелювання послідовного алгоритму вважаємо, що обчислювальна частина виразу (7) розпаралелюється відповідно до доступної паралельної структури реконфігурованої області, а сумарний час реконфігурації є послідовною частиною, розпаралелювання якої неможливо зважаючи на послідовні засоби реконфігурації та інтерфейс конфігурації. На підставі виразу (2) отримаємо наступний аналітичний вираз методу повторного використання апаратних ресурсів функціональних блоків для алгоритму обчислень, поданого графом ЯПФ:

$$T_G{}^{\text{МПВ}} = \sum_{j=1}^{K}\Delta T_j + \sum_{k=1}^{w}\max\{T_h \big| h = \overline{1, H_k}\}\,. \tag{8}$$

Потенційна можливість подальшого зменшення часу виконання паралельного алгоритму лежить в зменшенні послідовної компоненти часу, що може бути реалізовано шляхом розпаралелювання процесів обчислення та реконфігурації. Це можливо за застосування, наприклад, контролерів реконфігурації, що мають доступ до реконфігурованої структури без залучання загального комунікаційного середовища [4]. Надалі виконання реконфігурації за принципом «як можна раніше», переміщує всі процедури реконфігурації на попередній ярус, що за можливості розпаралелювання процесів реконфігурації та обчислення, призводить практично до повного знехтування часом реконфігурації і наближає до нуля непродуктивні витрати часу реконфігурації. Виключеннями є процедури реконфігурації першого ярусу графу алгоритму ЯПФ.

Відповідно до вищесказаного отримаємо математичну модель методу повторного використання апаратних ресурсів функціональних блоків з максимальним видаленням непродуктивної складової часу реконфігурації для макрографу, поданого в ЯПФ:

$$T_G{}^{\text{МПВ}} = \sum_{j=1}^{K}T_{comm_j} + \sum_{k=1}^{w}\max\Big(\sum_{h=1}^{H_k}\Delta T_h,\, \{T_h \big| h = \overline{1, H_k}\}\Big)\,. \tag{9}$$

де, послідовна компонента T_{comm} визначає послідовність процесів, які остаточно не можуть бути розділені у часі, наприклад, процеси обміну даними під час реконфігурації, що задіюють загальне комунікаційне середовище.

Апаратна реалізація засобів зменшення накладних видатків реконфігурції

Для реалізації запропонованого методу розроблено структуру обчислювального модуля, що вдосконалює відомі архітектури масштабованих реконфігурованих суперкомп’ютерів [1, 2, 4] шляхом додавання в його склад багаторівневої кеш-пам’яті (рис. 3). До складу реконфігурованого модуля додано спеціальний контролер реконфігурації, який дозволяє зменшити час реконфігурації

за рахунок розділення логічної і фізичної послідовності реконфігурації і розвантажування локального процесора [1, 4].

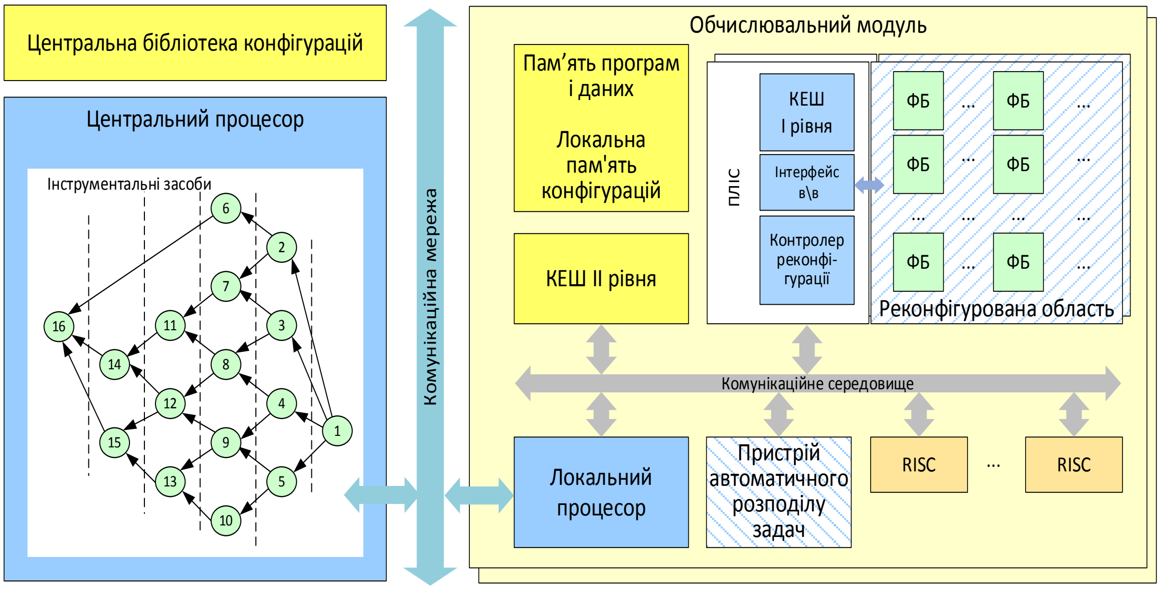


Рис.3. Структура обчислювального модуля

Для забезпечення механізмів зменшення непродуктивних комунікаційних витрат часу, розроблено новий спосіб збереження конфігураційних даних апаратних задач, детально описаний в роботі [4], за якого часто затребувані конфігурації апаратних задач зберігаються, в заздалегідь сконфігурованому вигляді, на поверхні реконфігурованої області ПЛІС. Окрім часового прискорення такий спосіб забезпечує зберігання обмежених ресурсів внутрішньої пам'яті ПЛІС. Апаратні задачі, що затребувані з найменшою частотою, зберігаються у вигляді конфігураційних файлів у зовнішній локальній пам'яті. Це дозволяє зменшити час повторного завантаження конфігураційних даних і зберегти обмежені апаратні ресурси мікросхеми ПЛІС. Багаторівнева швидкодіюча пам'ять конфігурацій, що реалізована засобами декількох прошарків кеш-пам'яті між центральною бібліотекою конфігурацій і реконфігурованою структурою мікросхеми ПЛІС, забезпечує ефективний облік та підтримку розкладу зберігання конфігурацій на поверхні реконфігурованої області. Кеш-прошарок першого рівня, реалізований на борту мікросхеми ПЛІС поряд з реконфігурованою структурою, застосовується для збереження слів стану функціональних блоків апаратних задач. Другий рівень розміщується в безпосередній близькості до мікросхеми ПЛІС, що дозволяє максимально мінімізувати комунікаційні витрати, наприклад у локальній зовнішній пам'яті конфігурацій обчислювального модуля. Кеш-пам'ять другого рівня застосовується для збереження слів стану і прискорення пошуку конфігураційних даних в локальній пам'яті конфігурації.

Застосування методу повторного використання апаратних ресурсів в реконфігурованих обчислювальних системах керованих потоком даних

Реалізація моделі обчислень, керованих потоком даних, окрім загального прискорення виконання паралельного обчислювального алгоритму, дозволяє

перенести керувальну складову процесу реконфігурації на структурний рівень обчислювального модуля, забезпечуючи тим самим зменшення складової часу управління $T_{control}$ (1). Особливість форматів слів команд та даних [6], за реалізації обчислень, керованих потоком даних, дозволяють забезпечити ефективну реалізацію принципу реконфігурації «як можна раніше». Чергова команда, що надходить на виконання, містить інформацію щодо набору вершин-нащадків графу алгоритму, для яких ця команда готує вхідні дані. Ця інформація надає можливість здійснювати реконфігурацію вершин-нащадків паралельно з виконанням обчислень відповідних батьківських вершин графу алгоритму.

Для керування обчисленнями в структуру обчислювального модуля реконфігурованої обчислювальної системи додано апаратний пристрій автоматичного розподілу задач, структура і принцип функціонування якого описані в роботі [6].

Граф алгоритму М-програми, який відповідає моделі обчислення, керованій потоком даних, зображений на рис. 4. Ущільнення часу виконання М-задач призводить до загального прискорення виконання алгоритму, у порівнянні з реалізацією графа, поданого в ЯПФ.

Запропонована математична модель (9) цілком відповідає моделі обчислень, керованих потоком даних. Для застосування запропонованого методу повторного використання апаратних ресурсів стає задача визначення послідовності найтриваліших задач для обрахування часу обчислення, для вирішення якої запропонований наступний спосіб визначення ефективного маршруту, що базується на знаходженні мінімальної, за сумарним часом виконання, послідовності задач. За геометричною інтерпретацією (рис. 4) ефективний маршрут будується на базі найтриваліших із набору готових до виконання задач. Це дозволяє видалити всі затримки на очікування вхідних даних.

Подамо вихідну М-програму макрографом $G_D = (N, D)$, де N – множина вершин, що відповідають макрозадачам, а D – множина ребер, що визначають відношення між макрозадачами (рис. 4).

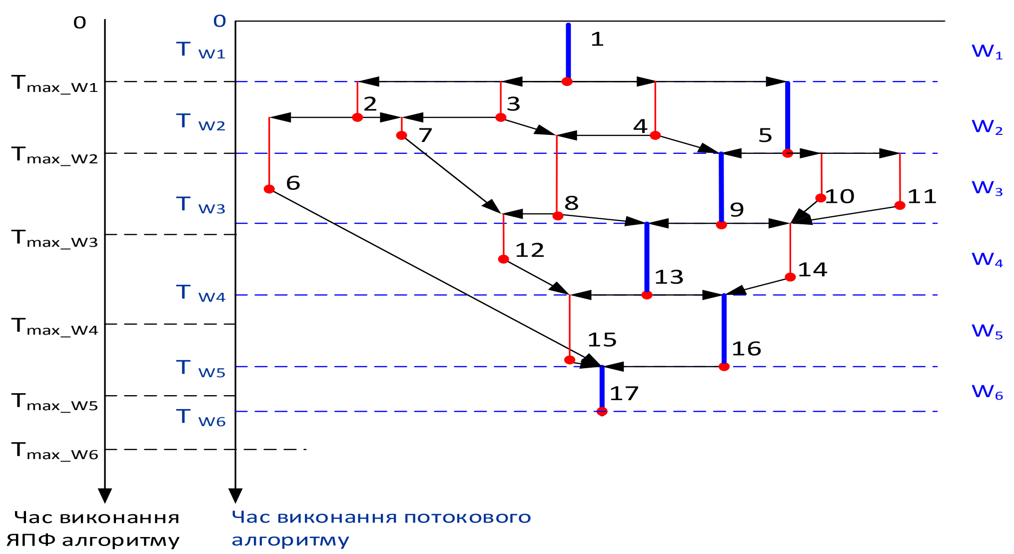


Рис. 4. Граф М-програми моделі обчислень керованих потоком даних

Опишемо граф програми в вигляді матриці зв'язності M_D (рис. 5) розмірності $(v \times v)$, де v – кількість вершин графу G_D. Рядки $M_D[g] \mid g = \overline{1,v}$ відповідають виконуваним задачам. У стовпцях $M_D[s] \mid s = \overline{1,v}$ кожного рядка $M_D[g]$ розміщується одиниця, якщо задача T_g готує дані для задачі T_s. Отримати множину всіх батьківських задач, які готують дані для задачі T_s, можна, проаналізувавши елементи стовбця $M_D[s]$, в якому індекси g вказують на шукані батьківські задачі T_g, якщо $M_D[g,s] = 1$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Рис. 5. Матриця зв'язності

Застосовуються наступні позначення: T_i – очікуваний час виконання i-ї задачі; $i = \overline{1,q}$, де q – кількість вершин-нащадків батьківської вершини T_g, $i_\max$ – індекс задачі, що має максимальний час виконання серед набору вершин-нащадків $Q_g = \{T_i \mid i = \overline{1,q}\}$; T_D – час виконання обчислювального алгоритму, обрахований за шуканим ефективним маршрутом. Далі наведено алгоритм пошуку ефективного маршруту:

Крок 1. Початкові встановлення: $g := 0, k := 0$

Крок 2. ($g{+}{+}, k{+}{+}$) Аналіз чергового рядку матриці $M_D[g]$ виконується за умови готовності результату задачі T_g відповідної вершини алгоритму, що відповідає наявності вхідних даних для всіх її вершин-нащадків. Формується масив задач-нащадків Q_g, для яких задача T_g підготувала вхідні дані, із якого обирається найтриваліша за часом виконання задача $T_{i_\max}$. Очікуваний час її виконання додається до загального часу виконання алгоритму $T_D := T_D + T_{i_\max}$.

Крок 3. Формування послідовності задач ефективного маршруту: $W[k] := i_\max$.

Крок 3. ($g = s$) Пошук рядку, для якого задача $T_g \mid g = s$ готує вхідні дані.

Крок 4. Якщо $M_D[g]$ не кінцева вершина графу: $M_D[g,s] \neq 1 \mid g = s$ (вихідна вершина алгоритму), повернення на крок 2.

Висновки

Запропонований метод повторного використання апаратних ресурсів функціональних блоків та апаратні засоби його реалізації, що дозволяють забезпечити інтенсивне прискорення реконфігурації за рахунок видалення всієї непродуктивної складової часу реконфігурації. Для забезпечення моделі обчислень, керованих потоком даних, запропоновано спосіб визначення часу виконання обчислювального алгоритму, що базується на знаходженні мінімального за часом виконання маршруту, позбавленого затримок очікування даних.

Література

1. Al-Wattar A. Efficient On-line Hardware/Software Task Scheduling for Dynamic Run-time Reconfigurable Systems / A. Al-Wattar, S. Areibi, F. Saffih // Proceeding in 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW). – 2012. – P. 401 – 406.
2. Ahmed W. Adaptive Resource Management for Simultaneous Multitasking in Mixed-Grained Reconfigurable Multi-core Processors /W. Ahmed, M. Shafique, L. Bauer, J. Henkel // Proceedings of the 9th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) (Taiwan, Taipei, 9 – 14 October 2011). – IEEE, 2011. – P. 365 – 374.
3. Liu S. Achieving Energy Efficiency through Runtime Partial Reconfiguration on Reconfigurable Systems / S. Liu, R.N. Pittman, A. Forin, J.-L. Gaudiot // Transactions on Embedded Computing Systems (TECS). – USA, NY, New York, ACM, 2013. – Vol. 12. – P. 72:1 – 72:21.
4. Кулаков Ю.О. Організація багаторівневої пам’яті в реконфігурованих обчислювальних системах // Ю.О.Кулаков, І.А.Клименко / Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. Наук. Пр. – К.: Век+, 2014. – №61. – С. 18 – 26.
5. Кулаков Ю.О. Метод оптимізації ярусно-паралельної форми подання задачі для реконфігурованих обчислювальних систем / Ю.О. Кулаков, І.А. Клименко // Електроніка та зв’язок. – К: НТТУ «КПІ», 2014. – Том 19, №4(81). – С. 90 – 96.
6. Жабин В.И. Архитектура вычислительных систем реального времени / В.И. Жабин. – К : ВЕК +, 2003. – 176 с.

Literatura

1. Al-Wattar, A., Areibi, S., Saffih, F. Efficient On-line Hardware/Software Task Scheduling for Dynamic Run-time Reconfigurable Systems // Proceeding of the 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), China, Shanghai, 21 – 25 May, 2012. – 2012. - pp. 401 – 406.
2. Ahmed W., Shafique, M., Bauer, L., Henkel, J. Adaptive Resource Management for Simultaneous Multitasking in Mixed-Grained Reconfigurable Multi-core Processors // Proceedings of the 9th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Taiwan, Taipei, 9-14 October, 2011. – 2011. – pp. 365 – 374.
3. Liu S., Pittman, R.N., Forin, A., Gaudiot J.-L. Achieving Energy Efficiency through Runtime Partial Reconfiguration on Reconfigurable Systems // Transactions on Embedded Computing Systems (TECS). – 2013. – Vol. 12. – pp. 72:1 – 72:21.
4. Kulakov Y.O., Klymenko I.A. The multilevel memory in the reconfigurable computing system // Visnyk NTUU «KPI». Informatyka, upravlinnia ta obchislyuvalna technika: Zb. nauk. pr. – K.: VEK+, 2014. – №61. – pp. 18 – 26. (in Ukrainian)
5. Kulakov Y.O., Klymenko I.A. The optimization method of a macro dataflow graph for reconfigurable computing systems // Electronika i zvjazok [Electronics and Communication]. – Vol. 19, № 4(81). – 2014. – pp. 90 – 96. (in Ukrainian)
6. Zhabin, V.I. The architecture of the real-time computer systems. – K.: VEK+, 2003. – 176 p. (in Russian)

RESUME

G.M. Luckiy, I.A. Klymenko

Reduced reconfiguration overhead in the reconfigurable computing systems

In this article, we propose new reconfiguration methods, that allow minimize reconfiguration overheads, providing performance improvements for dynamically reconfigurable computing systems.

Proposed method of the hardware resources reuse allows decrease nonproductive reconfiguration time expenses by decreasing communicational time component. Mathematical model of hardware resources reuse method prove, that proposed method for MDG presentation of computing algorithm of M-programs deletes total nonproductive reconfiguration time component, providing intensive reconfiguration acceleration. Obtained analytic expressions show, that amount of non-productive time component, using linear principle, depends from the number of repeated tasks. Thus, proposed method is the most effective for algorithms that contain a big amount of similar tasks.

Proposed hardware management tools for allocation scheduling and functional block configuration support, basing on the multilevel memory, provide effective support for methods of decreasing reconfiguration overheads, and allow effectively decrease computational complexity of the managing reconfiguration and to resolve problem of on-board memory limited resources.

Applying data-flow computational model increase effectiveness of dynamically reconfigurable systems, allowing besides overall performance acceleration of the computational algorithm, move management component of the reconfiguration process to the structural level of the computational module, decreasing computational complexity of the reconfiguration management. To use method for hardware resources reuse in dynamically reconfigurable systems, which are data-flow controls. The new way of determining execution time for data-flow program was proposed, which allows to get sequence of tasks. Their overall execution time equals to the minimum execution time of the computational algorithm.

Г.М. Луцький, І.А. Клименко

Зменшення накладних видатків реконфігурації в реконфігурованих обчислювальних системах

У статті запропоновано нові засоби реконфігурації, що дозволяють мінімізувати накладні видатки реконфігурації, забезпечуючи підвищення продуктивності динамічно реконфігурованих обчислювальних систем.

Запропонований метод повторного використання апаратних ресурсів дозволяє зменшити непродуктивні витрати часу реконфігурації за рахунок зменшення комунікаційної складової часу. Математична модель методу повторного використання апаратних ресурсів обґрунтовує, що запропонований метод для ЯПФ подання обчислювальних алгоритмів М-програм видаляє всю непродуктивну складову часу реконфігурації забезпечуючи інтенсивне прискорення реконфігурації. Отримані аналітичні вирази зображують, що обсяг непродуктивної складової часу за лінійним законом залежить від кількості повторюваних задач. Таким чином, застосування запропонованого методу найбільш ефективне в алгоритмах, що містять велику кількість однотипних задач.

Запропоновані апаратні засоби управління розкладом розміщення та підтримки конфігурацій функціональних блоків, на базі багаторівневої пам'яті, забезпечують ефективну підтримку методу зменшення накладних видатків реконфігурації та дозволяють зменшити обчислювальну складність алгоритмів управління реконфігурацією та вирішити проблему обмеженості ресурсів вбудованої пам'яті ПЛІС.

Застосування моделі обчислень керованих потоком даних підвищує ефективність динамічно реконфігурованих обчислювальних систем, дозволяючи окрім загального прискорення виконання обчислювального алгоритму, перенести управляючу складову процесу реконфігурації на структурний рівень обчислювального модуля, зменшуючи обчислювальну складність алгоритмів управління реконфігурацією. Для застосування методу повторного використання апаратних ресурсів в динамічно реконфігурованих системах, керованих потоком даних, запропонований новий спосіб визначення часу виконання потокової програми, який дозволяє отримати послідовність задач, сумарний час виконання яких відповідає мінімальному часу виконання обчислювального алгоритму.

Надійшла до редакції 28.05.2015

